Formalization of the class of problems solvable by a nondeterministic Turing machine

Anatoly D. Plotnikov*

Abstract

The objective of this article is to formalize the definition of NP problems.

We construct a mathematical model of discrete problems as independence systems with weighted elements. We introduce two auxiliary sets that characterize the solution of the problem: the adjoint set, which contains the elements from the original set none of which can be adjoined to the already chosen solution elements; and the residual set, in which every element can be adjoined to previously chosen solution elements.

In a problem without lookahead, every adjoint set can be generated by the solution algorithm effectively, in polynomial time.

The main result of the study is the assertion that the NP class is identical with the class of problems without lookahead. Hence it follows that if we fail to find an effective (polynomial-time) solution algorithm for a given problem, then we need to look for an alternative formulation of the problem in set of problems without lookahead.

1 Introduction

Solvability is the key problem in the theory of solution of discrete problems [1] – [7]. The input data and the solution result for any discrete problem are usually finite, and generally discrete problems do not suffer from the classical

*This paper is published in the journal "Cybernetics and Systems Analysis." Vol. 33, No. 5, 1997, pp. 635–640. Translated by Plenum Publishing Corporation, NY.

difficulty of total nonexistence of a solution algorithm. Many discrete problems have a trivial solution algorithm, which involves exhaustive enumeration of the elements of the solution. In practice, however, the trivial algorithm is inapplicable, because its computational complexity is too high even for a relatively small number of solution elements. Thus, if every discrete problem is interpreted as a problem of constructing a subset that satisfies given constraints among the elements of some initial n-set, then the trivial algorithm in general requires inspecting all 2^n subsets, which is obviously intractable. In such cases, we say that the trivial algorithm runs in exponential time, or has exponential complexity.

In the context of solvability of discrete problems we usually discuss the possibility of developing an algorithm that generates a solution in a time essentially shorter than the running time of the trivial algorithm. A discrete problem is regarded as effectively solvable if the running time of the solution algorithm is polynomial in the size of the problem. Such algorithms are called polynomial-time algorithms. Here and in what follows, the size of a discrete problem is the number n of elements in the input set.

The difficulties that arise in the process of development of solution algorithms for various discrete problems have led to the identification of a class of problems for which it is expedient to look for effective or polynomial-time algorithms. First, all problems for which no solution algorithm exists (e.g., solvability of polynomial equations in integers) or for which the number of solutions depends exponentially on the size of the problem (e.g., finding all 2^{n-2} covering trees of an n-graph) have been excluded [2, 3]. Among the set of discrete problems for which the number of solution elements (the length of the solution) is a polynomial function of problem size we focus on problems that are solvable by a nondeterministic Turing machine (NTM) in polynomial time. Discrete problems satisfying these constraints form the class NP.

Effective (polynomial-time) algorithms are available for solving some NP problems, and we accordingly identify a subclass P $\subset$ NP of problems with polynomial-time algorithms. For many practically important NP problems, however, attempts to find effective solution algorithms have failed.

The issue of discrete-problem solvability thus involves the relationship between the class NP and its subclass P. Some authors maintain that a strict inclusion applies, i.e., $P \subset$ NP and $P \neq$ NP, while others claim that P= NP. This disagreement among mathematicians is primarily due to ambiguously defined notions of NTM operation. The objective of this article is to formalize the definition of NP problems.

We consider the solution of an individual enumerative NP problem. By analyzing the operation of the NTM in the process of solving the problem, we establish that different interpretations of NTM operation lead to an ambiguous description of the process. In one interpretation, the NTM constructs the next intermediate solution using only the previously generated computation results; the other interpretation ignores this important feature.

We thus establish that unacceptably large enumeration during problem solving arises only when the NTM chooses the next solution element by inspecting an exponential number of all (final or support) solutions. The set of discrete problems is thus partitioned into two disjoint sets: problems for which the next solution element can be chosen without inspecting all the support solutions; and problems for which such choice is impossible. Problems of the first class are called problems without lookahead, while problems of the second class are called inherently exponential.

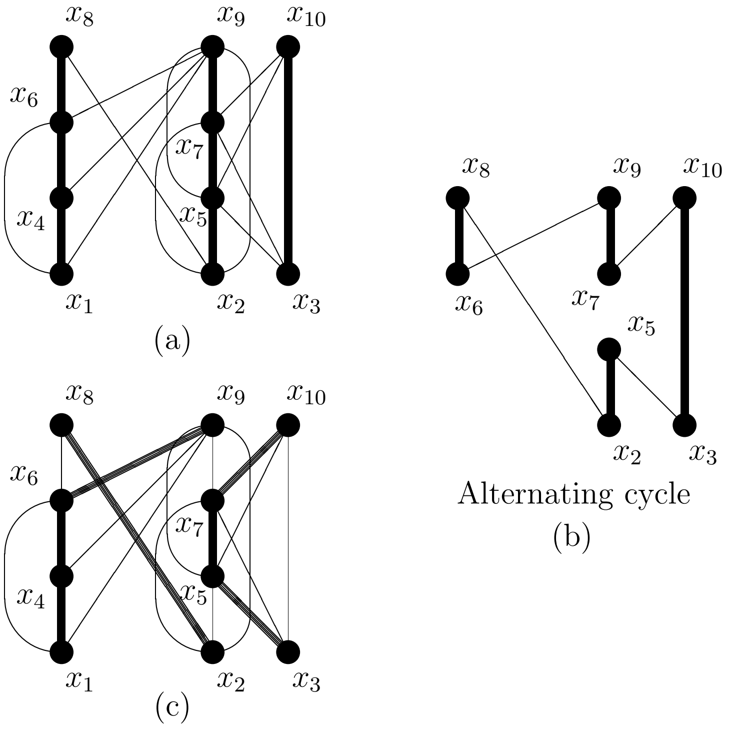
We construct a mathematical model of discrete problems as independence systems with weighted elements. We introduce two auxiliary sets that characterize the solution of the problem: the adjoint set, which contains the elements from the original set none of which can be adjoined to the already chosen solution elements; and the residual set, in which every element can be adjoined to previously chosen solution elements.

In a no-lookahead problem, every adjoint set can be generated by the solution algorithm effectively, in polynomial time.

The main result of the study is the assertion that the NP class is identical with the class of problems without lookahead. Hence it follows that if we fail to find an effective (polynomial-time) solution algorithm for a given problem, then we need to look for an alternative formulation of the problem in the set of no-lookahead problems.

2 Example of a discrete problem

Consider the acyclic digraph shown in Fig. 1 (a) (in all figures, the arcs are directed from bottom up). The transitive closure graph of the acyclic digraph is obviously a graph of a strict partial ordering. Any algorithm that constructs the maximum matching in a bipartite graph (see, e.g., [6]) can be applied to partition the nodes of this graph into a minimum number of chains (a so-called minimum chain partition, MCP). One of these MCPs contains the nodes and the arcs of the transitive closure graph that are shown by thick lines in Fig. 1 (a).



(a)

Alternating cycle
(b)

(c)

Fig. 1:

It is easy to see that the first chain in the MCP starting from the node x_1 contains two pairs of independent digraph nodes: x_1, x_8 and x_4, x_8. In general, the transitive closure graph can have several different MCPs, and the transition from one MCP to another is possible if we find an alternating cycle or an alternating chain.

Suppose that in the transitive closure graph for a given MCP it is required to find an alternating chain or an alternating cycle that takes us from the current MCP to another MCP such that none of the chains contains independent nodes of the original graph.

Is this an NP problem? The solution of this problem – an alternating cycle – obviously contains a number of elements that depends linearly on the number of nodes of the acyclic digraph. The problem is thus NP if it is solvable by NTM in polynomial time.

According to one interpretation, the NTM operates in two stages [3, 4, 6]: first the machine "guesses" some sequence of solution elements, and then it decides in polynomial time that the guessed sequence is a solution of the given problem. It is thus assumed that the guessing stage and the verification stage are both executed by the NTM in polynomial time. A key point in this interpretation is the feasibility of deciding in polynomial time that the solution is correct.

The correctness of the presented solution – an alternating cycle – obviously can be checked in polynomial time for our problem. Therefore, according to this interpretation, the problem is NP.

Yet there is also an alternative interpretation of NTM operation [1, 5, 7]. In [5], the operation of an NTM is illustrated by the problem of finding a correct k-coloring of some n-graph. There are k^n different ways to paint the nodes of an n-graph in k colors. It is required to decide if at least one of these colorings is a correct coloring. To this end it is obviously sufficient to examine all the edges of the colored graph, and if the end points of each edge are painted in different colors, then the coloring is correct.

The number of edges in a graph is of order $O(n^2)$. According to this NTM interpretation, we need to check simultaneously all k^n colorings, and the entire checking procedure is a linear function of the length of the input data – the number of elements in the adjacency matrix of the graph.

Under this interpretation, the main distinction between the operation of the NTM and the operation of a deterministic Turing machine (DTM) is that the NTM checks concurrently the correctness of all alternatives. Curiously, however, some authors (see, e.g., [1]) use both interpretations of NTM operation simultaneously.

Note that if we adopt the second interpretation of NTM operation, then the NTM goes from one state to the next only on the basis of previously generated computation results. This is also confirmed by simulating the operation of an NTM in a DTM with exhaustive enumeration of all computations [1]. We know that in each computation step the DTM goes from one state to another (and writes appropriate records on the output tape) only on the basis of previously generated (intermediate) results.

Let us consider from this point of view the construction of the alternating cycle shown in Fig. 1 (b). (Figure 1 (c) shows the MCP generated by this cycle.) Figure 2 (a) shows a part of this cycle, and Fig. 2 (b) shows the digraph with the elements of the partition of its transitive closure graph into chains generated by an intermediate computation result.

It is easy to see that the next thin arc (x_3, x_5) of the alternating cycle cannot be chosen unless we know in advance that the "thick" arc (x_7, x_9) will

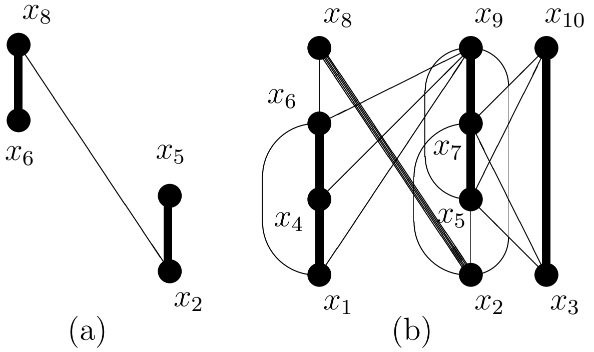


Fig. 2:

subsequently be included in the cycle being constructed. Thus, according to the second interpretation of NTM operation, this problem is not NP.

We have reached a diametrically opposite conclusion to the previous one. To eliminate the contradiction, we need to formalize the class of problems solvable by NTM in polynomial time.

The second interpretation of NTM operation is more appealing. Thus, in the example of checking the correctness of a graph coloring it is natural to assume that in each computation step the NTM decides which of the presented colorings are correct, and in the next step the decision about new correct colorings is reached by analyzing only the new edge in each "correct" option. If we adopt the other interpretation, then we have to agree that the NTM has an "instantaneous solver" that in each step allows the machine to "look ahead" into the required answer and thus decide which of the intermediate options is correct and which is not.

We have previously identified the "uninteresting" class of discrete problems that are inherently exponential. The problem of finding an alternating chain or cycle should be classified as inherently exponential, because in this problem we cannot use a partial (intermediate) result to pass to the next "correct" intermediate or final result. In general, this problem requires "inspecting" all final results (the number of which is an exponential function of the number of graph nodes).

In the next section we formalize the characteristics of such problems.

3 Set-theoretical model of discrete problems

In set-theoretical terms, many (finite) discrete problems involve selecting subsets π_i $(i = \overline{1,m})$ from some n-set R that satisfy given constraints [8]. A discrete problem is therefore defined as the 4-tuple $Z = (R.Q, M, f)$, where $R = \{r_1, \ldots, r_n\}$ $(n > 1)$ is called the *work set* and the *feasibility region* Q is a nonempty collection of subsets π of the set R that satisfy the following condition:

1^0) if $\pi \in Q$ and $\pi_1 \subset \pi$, then $\pi_1 \in Q$.

The set $M = \{\mu(r_1), \ldots, \mu(r_n)\}$ is a collection of nonnegative integers, and f is a function defined on Q. The pair (R, Q) is obviously an *independence system.*

Each element π in Q is called a feasible solution of problem Z, and the number $\mu(r_i) \in M$ $(\mu(r_i) > 0$ is the weight of the element $r_i \in R$ $(i = \overline{1,n})$.

In what follows we assume that for every $\pi \in Q$,

$$f(\pi) = \sum_{\forall r_i \in \pi} \mu(r_i).$$

The solution $\pi \in Q$ is called a *support solution* if there is no $\pi_1 \in Q$ such that $\pi \subset \pi_1$ and π is a proper subset of the set π_1. Denote by $B \subset Q$ the set of all support solutions of problem Z.

Problem Z is called *nontrivial* if $Card(B) = O(2^{p(n)})$. In other words, problem Z is nontrivial if the set of its support solutions contains exponentially many elements.

Suppose that it is required to find at least one support solution $\pi^* \in B$ such that $f(\pi^*)$ takes a specified value.

In a particular case, problem Z is called *extremal* if the function $f(\pi^*)$ takes an extremal value, i.e., for a maximization problem $f(\pi^*) \geq f(\pi)$ and for a minimization problem $f(\pi^*) \leq f(\pi)$, where $\pi \in Q$ is any support solution of problem Z.

4 Auxiliary solution sets

Denote by $W(\pi)$ the union of all feasible solutions $\pi_l \in Q$ of problem Z each of which includes the solution $\pi \in Q$,

$$W(\pi) = \bigcup_{\forall \pi_l \supseteq \pi} \pi_l.$$

Clearly, $W(\pi) \subset R$.

The set $S(\pi) = R \setminus W(\pi)$ is called *adjoint* to the solution $\pi \in Q$. Thus, an adjoint set consists of those and only those elements of the work set R whose union with the given feasible solution $\pi \in Q$ forms a subset of the set R that is not a solution of problem Z.

Theorem 1 *If $\pi_1, \pi_2 \in Q$ and $\pi_1 \subset \pi_2$, then $S(\pi_1) \subset S(\pi_2)$.*

Clearly, if $\pi_1 \subset \pi_2$, then $W(\pi_2) \subset W(\pi_1)$. Thus $S(\pi_1) = R \setminus W(\pi_1 \subset R \setminus W(\pi_2) = S(\pi_2.\circ$

The set $R(\pi) = R(\pi \cup S(\pi))$ is called the *residual set* for the solution $\pi \in Q$.

Theorem 2 *If $\pi \in Q$ and $r \in R(\pi) \neq \oslash$, then $\pi \cup \{r\} \in Q$.*

Indeed,

$$R(\pi) = R \setminus (\pi \cup S(\pi)) = (R \setminus \pi) \cap (R \setminus S(\pi)) =$$

$$= (R \setminus \pi) \cap W(\pi) = (R \cap W(\pi)) \setminus \pi = W(\pi) \setminus \pi.$$

Therefore, for $R(\pi) \neq \oslash$, the set $\pi \cup \{r\}$ is included in at least one feasible solution from Q and is thus also a solution by property 1^0.$\circ$

Theorem 3 *If $\pi \in Q$ is a support solution, then $R(\pi) = \oslash$ and $\pi \cup S(\pi) = R$.*

Let $\pi \in Q$ be a support solution of problem Z. Assume that $R(\pi) \neq \oslash$. This leads to the conclusion that the region Q contains a solution $\pi_1 = \pi \cup \{r\}$ ($r \in R(\pi)$), which properly includes the solution $\pi \in Q$. A contradiction with the definition of support solution.

The relationship $\pi \cup S(\pi) = R$ for every support solution $\pi \in Q$ follows from the definition of residual set when $R(\pi) = \oslash$.$\circ$

5 Class of problems without lookahead

Let T be the set of all possible problems Z. It follows from the above discussion that the issue of solvability of problem Z involves developing an algorithm (a deterministic Turing machine) that finds at least one support solution $\pi \in B$ of the problem for which the function $f(\pi)$ takes a specified value and does it in a time polynomial in the number of elements of the work set R.

In the set T we identify the subclass T_1 of problems in which the set of support solutions contains exponentially many elements (more precisely, the number of elements is a function of the form $2^{p(n)}$). Any problem $Z \in T_1$ is called *nontrivial.* In what follows we only consider the set of nontrivial problems $T_1 \subset T$ for which $Card(B) = O(2^{p(n)})$.

We say that the adjoint set $S(\pi)$ for a given solution π is determined *effectively* if for all elements $r_i \in R \setminus \pi$ we can decide in polynomial time the truth of the predicate "$\pi \cup \{r_1\} \in Q$" or the predicate "$\pi \cup \{r_1\} \in Q$".

Problem Z is called a *problem without lookahead* if for every feasible solution $\pi \in Q$ the adjoint set is determined effectively.

Theorem 4 *If $Z \in T_1$ is a no-lookahead problem, then it is solvable by a nondeterministic Turing machine in polynomial time.*

Indeed, by definition the size of problem Z is the number of elements n in the work set R, and the length of the solution, defined as the number of elements in some support solution $\pi \subset R$, is a linear function of n. Noting that Z is a no-lookahead problem, the NTM should compute all the feasible solutions simultaneously. Hence it follows that every problem $Z \in T_1$ is solvable by NTM in polynomial time.$\circ$

Theorem 5 *The class NP is identical with the class of no-lookahead problems, i.e., $T_1 = NP$.*

By Theorem 5, $T_1 \subset$ NP. By description, the class NP does not include inherently exponential problems. Thus, $T_1 =$ NP.$\circ$

6 Conclusion

At a first glance it would seem that the accepted interpretation of NTM essentially restricts the set of NP-complete problems that are considered when the interpretation allows solution "guessing." This is not so, however. We know that the formulation of a problem has an essential impact on the possibility of solving the problem. The accepted interpretation of NTM operation makes it possible to reject formulations that *a priori* require exhaustive enumeration of an exponential set of support solutions.

Thus, consider the problem to find the Hamiltonian cycle in a graph. This is an inherently exponential problem if it is formulated so that the construction of each feasible solution requires "guessing" a correct choice, i.e., "advance knowledge" of the collection of edges that forms a feasible solution, or belongs to at least one support solution – a Hamiltonian cycle.

The same problem can be formulated in a different setting: find a partition of the graph into a minimum number of cycles and edges. If the graph is Hamiltonian, then the solution of this problem produces the sought cycle.

References

[1] A. V. Aho. J. E. Hopcroft, and J. D. Ullman. The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading. Mass. (1974).

[2] E. N. Gordeev. "Choice problems and their solution," in: The Computer and Choice Problems [in Russian], Nauka, Moscow (1989). pp. 5–48.

[3] M. R. Garey and D. S. Johnson, Computers and Intractability, Freeman, San Francisco (1979).

[4] V. A. Emelichev and others. Lectures in Graph Theory [in Russian], Nauka, Moscow (1990).

[5] V. K. Leont'ev, "Combinatorics: past achievements and future prospects," in: The Computer and Choice Problems [in Russian], Nauka, Moscow (1989), pp. 49–88.

[6] C. H. Papadimitriou and K. Steiglitz, Combinatorial Optimization: Algorithms and Complexity, Prentice Hall, Englewood Cliffs, NJ (1982).

[7] E. M. Reingold, J. Nievergelt, and N. Deo, Combinatorial Algorithms: Theory and Practice, Prentice Hall, Englewood Cliffs, NJ (1977).

[8] A. D. Plotnikov, "Normal form of extremal combinatorial problems," Visnik VPI, 3(4), 73–76 (1994).